¿Para qué sirven los primos?

por
Beatriz Rubio Serrano
(Colaboradora del Instituto Universitario de Matemáticas y Aplicaciones)

A lo largo de nuestro paso por las diferentes etapas de la enseñanza, los números reales son los actores principales, y entre ellos los naturales, enteros y racionales son los protagonistas. Podría ser ese el motivo por el que la mayoría de nosotros creemos que son los únicos números espectadores de nuestro día a día. Poco o nada se nos cuenta de otro sistema numérico, actor secundario, que también está presente en nuestra vida cotidiana y que da consistencia a la trama de esta historia: El cuerpo finito de p elementos, $\{0,1,2,\ldots,p-1\}$ donde p es un número primo.

A estos cuerpos finitos se les han encontrado importantes aplicaciones, sobre todo en criptografía. La criptografía proviene de las palabras, *cripto* —ocultar— y *graphos* —escritura—. Podemos pensar que significa algo así como *el arte de ocultar mensajes*.

La criptografía tiene como objetivo principal enviar un mensaje de forma *oculta*, llamado *cifrado* o *encriptado*, y que sólo el receptor con una llave o clave secreta pueda descifrar y leer su contenido. A lo largo de la historia ha sido usada principalmente en guerras, medios de comunicación o agencias de seguridad nacional. En la actualidad es indispensable en muchos de los servicios que empleamos diariamente: Internet, teléfono, radio, televisión. O como ejemplo más concreto, cuando hacemos una compra online e introducimos nuestro número de tarjeta de crédito, este número se encripta usando la aritmética de módulos primos.

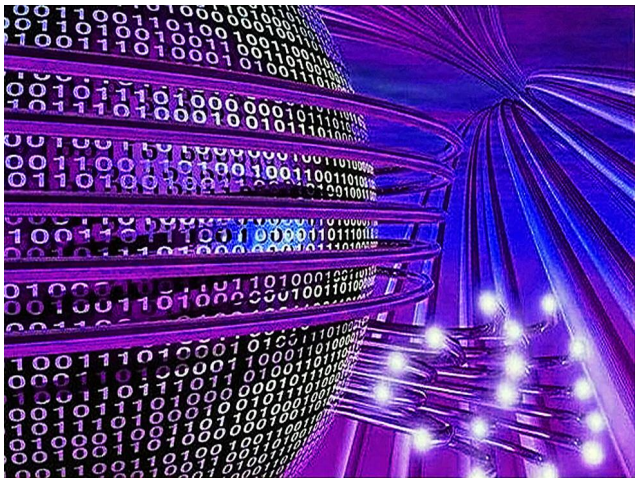
Para cada número primo p, existe un cuerpo finito con p elementos: $\{0,1,2,\ldots,p-1\}$. Estos p elementos comprenden un sistema numérico que es cerrado para la suma, resta, multiplicación y división módulo p. Obedecen las mismas reglas que las operaciones correspondientes con números racionales y reales.

Pero en este sistema numérico hay también algo especial. Si se toma cualquier elemento del cuerpo finito $\{0,1,2,\ldots,p-1\}$ y se eleva a la p-ésima potencia, en el sentido de la aritmética de módulo p, sorprendentemente se obtiene el mismo resultado. En otras palabras, $a^p \equiv a \pmod{p}$.

El *pequeño teorema de Fermat* es uno de los teoremas clásicos de teoría de números y su interés principal está en la aplicación de la primalidad y en criptografía. Hay otro teorema, análogo al pequeño teorema de Fermat pero con

aritmética módulo n, muy importante también en criptografía; el *teorema de Euler* o el *teorema de Euler-Fermat*.

$$a^{\varphi(n)} \equiv 1 \pmod{n},\ \forall n \in \mathbb{N} \text{ y } \operatorname{mcd}(a,n) = 1$$



Recordemos que la *función de Euler* $\varphi(n)$ es el número de números naturales entre 1 y $n-1$ que son primos entre sí con n; es decir, que no tienen divisores en común con n (aparte de 1).

Se demuestra exactamente de la misma manera que el pequeño teorema de Fermat. Probaremos el caso en que n es el producto de dos números primos; $n=pq$, con p y q primos y $p \neq q$.

En este caso, los números que no son primos entre sí con n son divisibles ya por p o por q. Los primeros tienen la forma p_i, con $i = 1, .., q-1$ (hay $q-1$ de ellos), y los segundos tienen la forma q_i, con $j = 1, .., p-1$ (hay $p-1$ de ellos). Por lo tanto, hallamos que $\varphi(n) = (n-1)-(q-1)-(p-1) = (p-1)(q-1)$

Por lo tanto, tenemos

$$a^{(p-1)(q-1)} \equiv 1 \pmod{pq},\ \forall a \text{ divisible por } p \text{ y } q$$
$$a^{1+m(p-1)(q-1)} \equiv a \pmod{pq},\ \text{es cierta } \forall a \in \mathbb{N} \text{ y } \forall m \in \mathbb{Z}$$

Esta ecuación es la base de uno de los algoritmos de encriptación más ampliamente utilizados, llamado algoritmo RSA (por Ron Rivest, Adi Shamir, y Leonard Adlmean, quienes lo describieron en 1977). La idea es que escogemos dos número primos p y q (existen varios algoritmos para generarlos) y decimos que n es el producto pq. El número n se hace público pero los números primos p y q no. Acto seguido escogemos un número e primo entre sí con $(p-1)(q-1)$. Este número también se hace público.

El proceso de encriptado convierte todo número a (como el número de una tarjeta de crédito) a a^e módulo n:

$$a \to b \equiv a^e \pmod{n}.$$

Hallamos un número d entre 1 y $(p-1)(q-1)$ tal que $de) \equiv 1 \pmod{(p-1)(q-1)}$ es decir, $de) \equiv 1 + m(p\text{-}1)(q\text{-}1)$, para algún $m \in \mathbb{N}$.

De esta forma:

$$a^{de} \pmod{n} \equiv a^{1+m(p-1)(q-1)} \pmod{n)} \equiv a \pmod{n}$$
por la fórmula anterior.

Por lo tanto, dado que $b = a^e$, podemos recuperar el número original a como sigue:

$$b \to b^d \pmod{n}$$

Por lo que hacemos públicos los números n y e, pero mantenemos en secreto d.

La fórmula $a \to b) \equiv a^e \pmod{n)}$ nos da la encriptación. Cualquiera puede hacerlo porque e y n son de dominio público.

La descripción nos da la fórmula $b \to b^d \pmod{n)}$, aplicada a a^e, nos devuelve el número a original y solo quienes conocen d pueden hacer esto.

La razón por la que este es un buen código de encriptación es que a fin de hallar d, que nos permite reconstruir los números codificados, hemos de conocer el valor de $(p-1)(q-1)$. Para ello tenemos que saber el valor de p y q, dos números primos divisores de n. Y estos se mantienen en secreto. Con un número n suficientemente grande, pueden tardarse muchos meses, incluso con una red de ordenadores potentes, en hallar p y q. La multiplicación de dos números primos grandes es una tarea fácil para un ordenador pero el procedimiento inverso de encontrar los dos números primos originales a partir de su producto puede resultar muy difícil. En 2009, un grupo de investigadores, empleando cientos de ordenadores en paralelo, tardaron dos años para factorizar en números primos un número de 232 dígitos.